Can language models understand visual information?

Anonymous ACL submission

Abstract

We introduce a novel benchmark for evaluating whether language models can recognize visual concepts through Scalable Vector Graphics (SVG) representations. SVG is a wide-spread format that encodes images as sequences of elementary geometric shapes with text-based markup, making visual information accessible to language models without requiring additional training or multi-modal architectures. This enables us to investigate a fundamental question: Do models trained on text develop an understanding of the visual properties – shapes, proportions, and colors – of real-world objects? To address this question, we present a benchmark consisting of minimal manually-created SVG depictions of common objects, and compare our results to a human-annotated baseline.

1 Introduction

While humans learn language in a multi-modal environment where words are grounded in perceptual experience, language models acquire their capabilities passively from mostly textual data. This raises a critical question: *Can language models develop an understanding of visual properties – shapes, proportions, colors, and spatial relationships – from linguistic data?*

Scalable Vector Graphics (SVG) format allows us to empirically tackle this question. Unlike raster images that encode visual information in pixel arrays, the vector SVG format represents images as *structured text* describing geometric primitives: circles, rectangles, paths, and their properties. This textual encoding makes visual content directly accessible to language models without requiring specialized vision components or multi-modal architectures. When a language model processes an SVG description of a yellow circle above a green rectangle, can it recognize this as a representation of the sun over grass?

We introduce a benchmark designed to systematically evaluate how much language models understand visual information. Our approach leverages the fact that many common objects can be abstracted into simple geometric compositions (for example Figure 1). By presenting models with minimal SVG representations and asking them to classify the depicted objects, we can probe whether training on text corpora has led to implicit learning of visual-spatial relationships.

Our benchmark consists of manually created, minimal SVG depictions of common objects, each paired with four manually-created plausible captions. By focusing on simple, compositional representations using basic shapes, we ensure that success on this task genuinely reflects understanding of how geometric primitives combine to represent objects, rather than pattern matching against memorized complex SVG structures. This design choice also enables systematic analysis of which visual concepts are well-grounded in language models and which remain challenging, providing insights for both understanding current models and improving future systems.

2 Related Work

Grounding in language models The question of whether language models can develop grounded representations of the visual world from text alone has been the subject of intense debate. Bender and Koller (2020) famously argued through their *octopus test* that systems trained solely on linguistic data cannot learn meaning without access to the communicative intent and real-world referents that ground language. This position, similar to earlier arguments such as Harnad's symbol grounding problem (Harnad, 1990), suggests that true understanding requires sensorimotor experience. However, recent empirical work challenges this view. Sharma et al. (2024) demonstrated that text-only

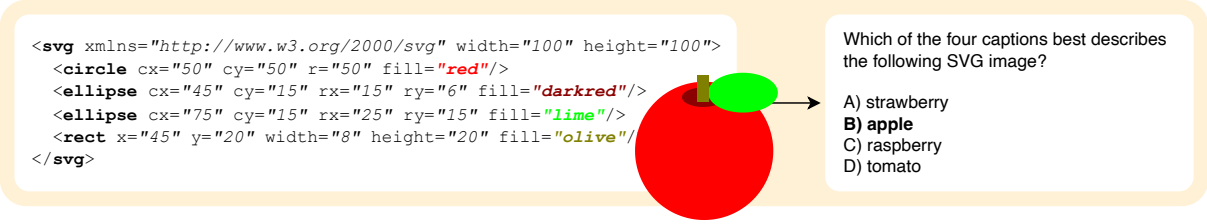

Figure 1: **An illustrative sample from our dataset.** The dataset consists of manually created minimal SVG representations of basic objects, coupled with four manually-written plausible captions where only a single one is correct.

language models possess sufficient visual knowledge to generate code that renders recognizable images. Similarly, Søgaard (2023) argues that referential grounding may be learnable from raw text alone, as linguistic patterns encode rich information about the structure of the world. The debate has taken new directions with the advent of multimodal models, though Yang et al. (2025b) surprisingly found that vision-language models do not consistently outperform text-only models on embodied knowledge tasks, suggesting that visual grounding may not provide the expected advantages. Our work contributes to this discussion by directly testing whether models can understand the textual descriptions of geometric primitives (in SVG format) to recognize visual objects, providing a controlled way of evaluating the grounding capabilities that emerge from different training regimes.

Existing SVG benchmarks In a recent survey paper, Malashenko et al. (2025) identify several existing benchmarks for SVGs. They separate the identified benchmarks into generation, editing, and understanding, where editing is the capability of a model to perform specific edits on an SVG. For example, Cai et al. (2024) use a simple and automatically generated dataset to evaluate the ability to answer questions, such as the relationship between several colored shapes. Zou et al. (2024) introduce VGBench, a benchmark for both generation and understanding. The latter part of the benchmark, Vector Graphics Question Answering (VGQA), contains questions about color, categories and usage. Finally, SGP-Bench, introduced by Qiu et al. (2025), tests a model's question answering capabilities, for example in identifying the contents of a given SVG given four question alternatives. Malashenko et al. (2025) note that VGBench and SGPBench both contain annotation errors due to how they were largely automatically generated and sourced from web-scraped data.

Data Gathering SVGs can be obtained in several ways. The easiest and most popular approach is to scrape images from the web – used in all aforementioned benchmarks, Cai et al. (2024), Zou et al. (2024) as well as Qiu et al. (2025). This approach not only risks data contamination with web-base pretraining datasets of language models, but these images are often of varying styles, made mostly of complicated Bézier curves, of uncertain quality, with potential copyright issues. and naturally often consist of logos and emojis. It also makes it difficult to have a consistent dataset for exploring, to which one can add more complexity in a controlled way, as it is more difficult to manipulate complex paths with many points, and styles might differ significantly, making small changes difficult to detect.

Our approach substantially differs from the previous work, because we insist on creating and annotating the images manually to ensure that they are minimal, high-quality and fully open.

3 Dataset

We present a new dataset of 200 images composed of elementary shapes. The dataset is manually drawn by two contributors that were asked to follow constraint guidelines to maximize simplicity of the images. All images are paired with a general description. Our dataset is unique in that although simple, it is easily compositional, and due to the images not being restricted to icons or emojis, it is easier to combine them with minimal effort to expand the dataset. As the project is open, it is possible to combine the images in many ways.

The dataset is mainly made using Inkscape, an open-source tool for creating vector graphics. The images are made in a 'cartoonish' style, and are not meant to be realistic; rather focusing on abstracting away from details and capturing the defining components of each object (Figure 1). The dataset is meant to represent various common themes, but

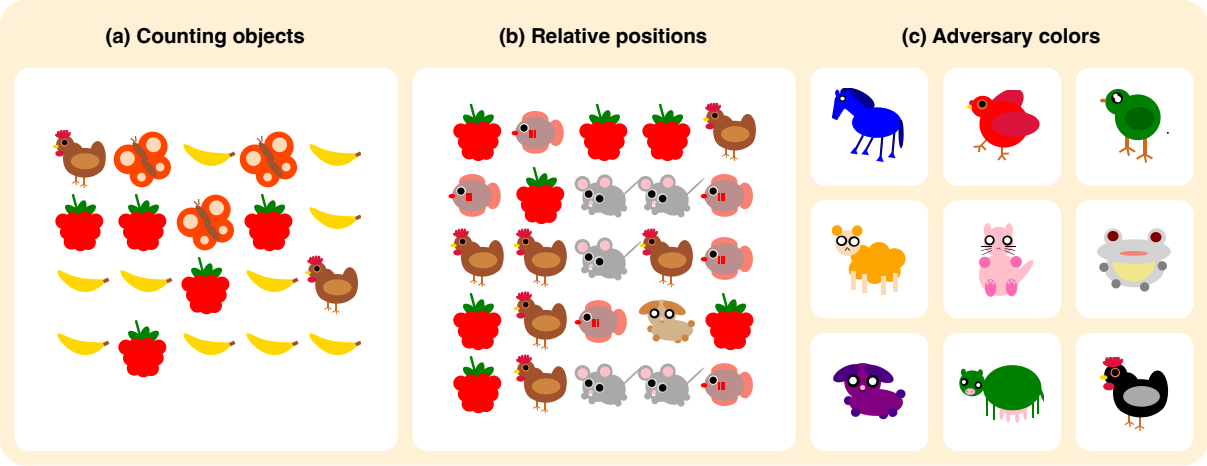

Figure 2: **Automatic expansion of the basic dataset.** The high-level abstraction of the SVG format allows for programmatic manipulation of the SVG images. We use this property for three additional experiments that test specific image-understanding aspects: counting objects (a), relative positioning (b) and color bias (c).

the initial thematic focus was on animals and food. While the focus has been on basic concepts, there are also some examples of more complicated images showing complex verbal actions.

The dataset is composed with two main principles in mind. First of all, the aim has been to keep the graphics simple, so the contributors followed a simple set of guidelines, such as restricting shapes to ellipses, circles, squares, rectangles and polygons. Secondly, the dataset is created with a degree of future compositionality in mind, in the sense that it should be easy to combine existing SVGs to represent more advanced or compositional concepts. An example of this is having an apple and a cat, allowing for SVGs representing various interactions between the cat and apple (picking, eating, etc.). We have to some degree experimented with possible compositionality in the dataset, and while most SVG are simple, the complete dataset contains some composed images, serving both as examples of more complex images and as further inspiration for dataset expansion.

Normalization The SVG format is complex and the images can in principle be made of nested matrix transformations and cross-referenced definitions. Since our goal is to test grounding, not arithmetics and syntax, we normalize each SVG to the minimal possible form (Figure 1). The resulting images are made of a flat sequence of basic elements, all coordinates are quantized and no transformations except for simple rotation are allowed. The colors are reduced to a limited set of 48 CSS colors that are presented in their word-form (for example `"red"` instead of `"#FF0000"`). We provide both the original and the normalized versions in the published dataset, but only the normalized files are used in the experiments.

Figure 3: **A subset of the dataset.** The images depict both basic objects (strawberry), as well as complex composite situations (a witch on a broom with a smartphone).

3.1 Automatic dataset expansion

In addition to seeing if models can identify SVG images, we also want to probe more specific aspects of visual understanding – whether language models can recognize the correct number of items, recognize various spatial positions, and variation in coloration; inspired by the previous research outlined above. All image samples for this dataset expansion were made with a subset of basic images (from the manually created dataset), and composed automatically using dedicated script that will be published alongside the dataset.

Counting In order to create a challenging dataset for encoding long-range context, we create composite image where a main object is duplicated 1–9 times in a 5×4 grid setup, with three other randomly selected objects serving as confounding elements. The number of confounding SVGs serves as the base for the incorrect question alternatives. We avoid using positional transformations, but rather focus on having each element in its place without being relative to the original item. The prompt replies are generated automatically. In Figure 2(a) we see an example of this setup.

Relative positioning In this setup, we focus on models' ability to identify objects *above*, *to the right of*, *below*, and *to the left of* a certain other object. The variations and labels were created automatically, from a subset of the main dataset. We employ a 5×5 grid setup, where the main object is randomly placed into one of the inner 3×3 positions. Then four other objects are randomly selected and placed either directly above, below, to the right, or to the left of the main object. The main image is always unique, and each direction has a unique secondary image, while the remainder of the grid is randomly filled with the secondary images. One of the resulting images is shown in Figure 2(b) where a dog serves as the main object.

Color For the last setup, we want to investigate the models' sensitivity to color information. Do the models still recognize a horse even when it is blue – or do they confuse it with other objects that are naturally blue? We identify parts of objects that can be colored differently and then programmatically alter the colors. Then we manually write three adversary labels for each color to test the robustness of models' visual understanding. Nine examples of the altered images are presented in 2(c), where the main color-giving areas are changed. For this subset, only a set of images that would require one or two colors to be changed were selected.

4 Experiments

This section describes the experimental methodology for the evaluation of image-recognition abilities of language models.

The main experiment In order to test how good language models are in recognizing images from descriptions of their constitutive shapes, we evaluate them on image recognition consisting of 200 SVG-caption pairs. Each image is paired with 4 plausible captions (manually created) with only a single correct caption (Figure 1).

Ablation experiments Next, we alter some aspects of the main experiment (turning all colors black, randomizing all coordinates or disabling chain-of-thought reasoning) to better understand what helps language models to recognize images – is that because they simply match the color names with the correct caption, or because they understand the spatial properties of the image?

Additional focused experiments As described in Section 3.1 and Figure 2, the SVG format allows us to easily manipulate different properties of the images. Therefore we can construct additional experiments that focus on certain aspects of visual understanding – we test if the models are able to count objects in an image, if they understand relative positioning and how robust they are to adversarially colored images.

Comparison of language and visual-language models Evaluating purely textual language models is difficult because modern models are usually natively multi-model. We thus conduct a focus experiment where we compare models trained only on text with the same models further trained on multimodal data. This will help us answer whether models can obtain visual understanding purely from text, or whether their performance should be attributed to multi-modal training. We use two model families with known training details for this experiment: Qwen3-VL (Bai et al., 2025a) and GLM-5V (Team et al., 2026).

4.1 Task setup

We opted for a multiple-choice question-answering setup for evaluating the language models. Compared to open-ended generation, this formulation allows for simple and accurate measurement of prediction accuracy. We specifically chose to manually curate the multiple choices to make them more challenging – compared to randomly selecting the choices from other captions. All tasks use the same setup and the same prompt template given below in Figure 4, and their validity was checked in a human baseline test, see Section 5.

4.2 Evaluated models

We evaluated 50 state-of-the-art closed- and open-weight language models available as of writing

```
Which of the four captions best describes the
following SVG image? Respond only with [[A]], [[B]],
[[C]] or [[D]].

## Image
```svg
{image}
```

Captions
A) {caption_1}
B) {caption_2}
C) {caption_3}
D) {caption_4}
```

Figure 4: **Prompt template.** We use this simple image-classification template for all experiments in this paper.

this article. As detailed in further experiments below, we focused on models that support chain-of-thought reasoning, which turned out to be crucial for achieving non-trivial performance on this task.

Closed-weight models These include the Claude 4 family of models (Claude Haiku 4.5, Claude Sonnet 4.6, and Claude Opus 4.7; Anthropic, 2025, 2026b,a), the GPT-5 family (GPT-5.4 nano, GPT-5.4 mini, and GPT-5.5; OpenAI, 2026a,b), the Gemini 3 family (Gemini 3 Flash, Gemini 3.1 Flash-Lite, Gemini 3.5 Flash, and Gemini 3.1 Pro; Google DeepMind, 2025), the Grok 4 family (Grok 4.1 Fast and Grok 4.3; xAI, 2025, 2026), Alibaba's proprietary Qwen models (Qwen3.6 Plus, Qwen3.6 Max Preview, and Qwen3.7 Max; Qwen Team, 2026b), and ByteDance Seed (Seed 1.6 and Seed 1.6 Flash; ByteDance Seed, 2025).

Open-weight models The models with publicly available checkpoints include: DeepSeek models (DeepSeek V3, DeepSeek V4 Flash, and DeepSeek V4 Pro; Liu et al., 2025; DeepSeek-AI, 2026), the Qwen3.5 series (Qwen3.5-9B, Qwen3.5-122B-A10B, and Qwen3.5-397B-A17B; Yang et al., 2025a; Qwen Team, 2026a) and the Qwen3.6 open-weight series (Qwen3.6-27B and Qwen3.6-35B-A3B; Qwen Team, 2026b), Z.ai's GLM models (GLM-4.7 Flash, GLM-5 Turbo, GLM-5V Turbo, and GLM-5.1; Zeng et al., 2025; Z.ai, 2026), NVIDIA's Nemotron family (Llama-3.3-Nemotron-Super-49B-v1.5, Nemotron-3-Nano-30B-A3B, and Nemotron-3-Super-120B-A12B; Bercovich et al., 2025; NVIDIA, 2025), the Hermes 4 models (Hermes 4 70B and Hermes 4 405B; Teknium et al., 2025), Kimi K2.6 (Bai et al., 2025b; Moonshot AI, 2026), the GPT-OSS models (GPT-OSS 20B and GPT-OSS 120B; Agarwal et al., 2025), the Gemma 4 family (Gemma 4 26B-A4B and Gemma 4 31B; Google DeepMind, 2026), Mistral models (Ministral 3 (3B, 8B, and 14B), Mistral Small 4, and Mistral Medium 3.5; Liu et al., 2026; Mistral AI, 2026a,b), MiniMax M2.7 (MiniMax, 2026), Xiaomi's MiMo (MiMo-V2 Flash, MiMo-V2 Omni, MiMo-V2.5, and MiMo-V2.5 Pro; LLM-Core Xiaomi, 2025; Xiaomi, 2026), and IBM Granite 4.1 8B (IBM Research, 2026).

5 Results

Human Baseline As a human baseline, 7 annotators volunteered to perform the same main task of choosing the correct answer (using the same prompt), giving an average score of 98% (sd of 1 percentage point). No annotators were native English speakers, but were allowed allowed to use a dictionary in case of unknown vocabulary. One common mistake is likely due to unfamiliar lexemes, as 4 annotators mislabeled *bilberry*. The remaining errors showed no clear pattern.

Main results The results of the main image-classification experiment are plotted in Figure 5 and detailed in Table 4. We can see that most of the 'frontier' large language models fail to reliably recognize the simple images, with accuracy below 80%. However, four of the evaluated models — Gemini 3 models and GPT 5.5 – achieved noticeably stronger results not far away from the 98% human baseline. The best open-weight model is Kimi K2.6 81.5%, reaching accuracy of just 61.5%.

In order to better understand the capabilities of current language models, Figure 5 shows the trade-off between real-world per-sample cost and accuracy. Since it is impossible to know the computational complexity of closed models, the per-sample cost on the open market (at `https://openrouter.ai/`) is the best approximation for the actual computation complexity of each model. Next, we use this data to estimate the cost-performance Pareto frontier in Figure 5, and use six models that lie near the frontier[1] – Gemini 3 Flash, GPT 5.5, Kimi K2.6, Claude Sonnet 4.6, Gemma 4 31B and DeepSeek V3 – for the further experiments below.

Ablation The results of the more detailed experiments on the frontier models are given in Table 1. The first ablation experiment shows that the best-performing models can lose more than 20 percentage points of accuracy when they are unable to

[1]They have to lie near the frontier and be able to turn off their reasoning ability, as explained in the experiment.

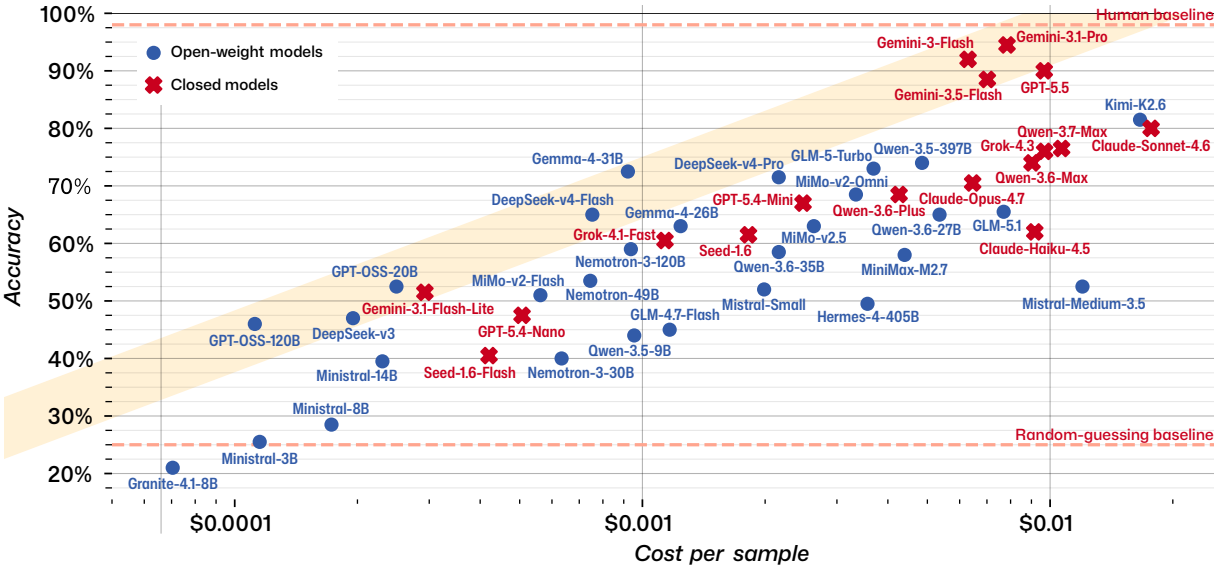

Figure 5: **Trade-off between computational cost and classification accuracy.** Models with openly available weights are plotted in blue and closed models are plotted in red. As the architectural details of most closed models are unknown, we use the the average cost of classifying a single sample as the approximation for the required floating-point operations (FLOPs). The Pareto-frontier of best-performing models is highlighted in yellow. We also indicate the human baseline and the random-guessing baseline as red dotted lines.

perform detailed chain-of-thought analysis of the images. The second ablation, where we turn all SVG elements black, does not lead to such a dramatic loss of performance. Especially compared to the third ablation – losing the positional information leads to a very clear drop highlighting the importance of this information.

Additional focused experiments Table 2 provides the results of the next group of experiments. Interestingly, when it comes to object counting, GPT 5.5 is able to perform this task perfectly, while other two strong models, Kimi and DeepSeek, struggle at this task. When it comes to adversary colors, most of the models show very robust understanding of the visual information and are not misguided to misclassify based on an unexpected color. DeepSeek seems more susceptible to adversary coloring, but it is also the worst-performing model on the main task (with 47% accuracy).

Comparison of language and visual-language models To investigate whether the recognition ability primarily emerges from text-only pretraining or rather from joint multi-modal training, we compare four pairs of language models with their visual-language counterparts in Table 3. Across all four pairs, the accuracy difference between the text-only and visual-language variant is statistically insignificant. For the three Qwen3 pairs, the visual-language variant scores marginally higher than the text-only one, while for GLM-5 the text-only variant scores slightly higher than GLM-5V.

6 Discussion

Our results provide some insights into the grounding of modern language models. The fact that the state-of-the-art systems are able to surpass 90% accuracy on minimal geometric representations of common every-day objects reveals they are able bridge symbolic and visual domains. Doing so consistently is still challenging, as evident from the long tail of under-performing systems, but it is clearly possible.

Multi-modal advantage The best-performing models on our benchmark – Gemini 3 models and GPT 5.5 – closely align with the 'frontier' of general-purpose language models. Which also means that all of these models actually multi-modal systems trained on both text and images (even though further details beyond that are unknown). One explanation of their success could thus be the grounding signal provided by the multi-modal training. However, their advantage is less pronounced than might be expected. When viewed through the lens of computational efficiency (Figure 5), the multi-modal models do not form a distinct performance class separate from text-only systems – the

Model	full scores	without reasoning	without colors	randomized coordinates
Gemini 3 Flash	92.0	78.5 (–13.5)	80.5 (–11.5)	38.5 (–53.5)
GPT 5.5	90.0	79.5 (–10.5)	86.0 (–4.0)	36.5 (–53.5)
Kimi K2.6	81.5	57.5 (–24.0)	68.5 (–13.0)	39.5 (–42.0)
Claude Sonnet 4.6	80.0	71.0 (–9.0)	70.5 (–9.5)	39.5 (–40.5)
Gemma 4 31B	72.5	60.5 (–12.0)	66.5 (–6.0)	38.0 (–34.5)
DeepSeek v3	47.0	44.0 (–7.0)	45.0 (–2.0)	33.0 (–14.0)

Table 1: **Ablation study of selected language models.** The models are evaluated: (i) without any chain-of-thought reasoning, (ii) on SVG images without any color information, and (iii) on SVG images with randomized coordinates. The table gives the absolute accuracy scores with the relative change (decrease) of performance in parentheses.

Model	object counting	positional relations	adversary colors
Gemini 3 Flash	87.0	94.4	97.4
GPT 5.5	100.0	100.0	99.4
Kimi K2.6	39.5	100.0	96.7
Claude Sonnet 4.6	85.2	88.8	89.6
Gemma 4 31B	67.9	50.0	89.0
DeepSeek v3	29.0	50.0	57.8

Table 2: **Additional focused experiments on the automatically-expanded datasets.** The table gives the accuracy scores (in %) for each task.

base model	text-only model	visual-language model
GLM-5-Turbo	73.0	71.0
Qwen3-235B-A22B	57.5	60.5
Qwen3-30B-A3B	46.0	47.0
Qwen3-8B	36.0	37.5

Table 3: **Comparison of language and visual-language models.** We provide the accuracy (%) of text-only vs. visual-language model variants on the main benchmark.

Pareto frontier includes both multi-modal and text-only models, with GPT-OSS and DeepSeek v3 and v4 (text-only architectures) achieving competitive cost-performance ratios. The focused comparative experiment in Table 3 provides further empirical evidence that multi-modal training alone cannot explain the observed level of visual understanding – language models can learn this ability from purely textual data.

Chain-of-thought as visual reasoning The dramatic performance degradation when chain-of-thought reasoning is disabled – up to 25 percentage points for Kimi K2.6 – reveals that these models are not performing simple pattern matching but rather engaging in compositional visual reasoning. The models appear to explicitly decompose the SVG representations into their constituent shapes, analyze spatial relationships, and then synthesize this information to identify objects.

Shape-based representations over color cues Perhaps most revealing is the relative impact of our ablations on model performance. While removing color information causes a moderate performance drop (2–13 percentage points), eliminating positional information devastates accuracy across all models. This 'shape preference' mirrors the findings of developmental psychology about how humans acquire object concepts, where spatial configuration and shapes are more fundamental than surface features (Landau et al., 1988). The robustness to adversarial coloring (Table 2) further confirms that models have learned to prioritize structural over superficial features.

Implications for grounding These findings suggest that language models trained on vast text corpora do develop some form of grounded representation and show signs of deep visual understanding. The models appear to have learned statistical associations between textual descriptions of spatial arrangements and object identities, enabling them to perform a form of 'symbolic visual reasoning' when presented with SVG markup. Still, the statistically-significant gap between the average human performance and the best model performance on this simple task indicates that this grounding remains incomplete. The models still lack the robust compositional understanding that allows humans to effortlessly recognize objects from minimal geometric representations.

The success of chain-of-thought reasoning in improving performance points to a potential path forward: rather than simply scaling model size or training data, developing architectures that better support compositional reasoning about spatial relationships may be key to achieving more human-like

grounding.

Analysis of output We look at the model output both to understand the difficulty of each image given the model lineup, and to understand the capabilities of the different models. To investigate the former, we looked at model prediction spread across the possible options. Some images were almost always predicted correctly by all models, such as *banana tree with fruit*, which all models identified correctly, while some had a limited but split distributions, as with 'an otter in the water, which had 26 correct predictions, but also 21 predictions for *a platypus in the water*. On the other hand are images that have relatively even distribution among their guesses, like *doughnut*, which had the following predictions: *hamburger*: 5, *confetti*: 18, *doughnut*: 14 and cupcake: 10. Looking at the top five performing models for the main task, only one image is mistaken by all models: *a steak with potatoes and carrots on a plate*, but these models also mistake simpler images such as *spoon*, and while many of the errors for these are complex, it is not completely clear-cut, neither in terms of which images are more difficult, or in terms of what makes certain images difficult for certain models.

7 Conclusion

We introduced a benchmark for evaluating the ability of language models to recognize objects from minimal SVG representations, addressing the fundamental question of whether language models develop grounded visual understanding. Our experiments reveal that even state-of-the-art models achieve limited accuracy on these simple geometric compositions, with performance critically dependent on spatial information and chain-of-thought reasoning rather than surface features like color. While models demonstrate some capacity for compositional visual reasoning through explicit decomposition of shapes and spatial relationships, the substantial gap between human and model performance on these minimal representations indicates that true visual grounding remains an open challenge. Our benchmark provides a simple yet effective diagnostic tool for measuring progress towards genuinely grounded AI systems.

Limitations

Using a simple dataset allows us create a starting point for SVG-based evaluation, while acknowledging that there are many aspects of SVGs that are not captured by our experiments. We also do not control for effects in the data, and to what extent the models have seen SVG or SVG-like data prior to the experiments. Later research should look into which elements contribute to understanding, and which are more problematic, especially as the image representations scale in complexity.

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, and 1 others. 2025. GPT-OSS-120b & GPT-OSS-20b model card. *Preprint*, arXiv:2508.10925.

Anthropic. 2025. Claude Haiku 4.5. https://www.anthropic.com/news/claude-haiku-4-5. Released 15 October 2025.

Anthropic. 2026a. Introducing Claude Opus 4.7. https://www.anthropic.com/news/claude-opus-4-7. Released 16 April 2026.

Anthropic. 2026b. Introducing Claude Sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6. Released 17 February 2026.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, and 1 others. 2025b. Kimi K2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, Ido Shahaf, Oren Tropp, Ehud Karpas, Ran Zilberstein, Jiaqi Zeng, Soumye Singhal, and 1 others. 2025. Llama-Nemotron: Efficient reasoning models. *Preprint*, arXiv:2505.00949.

ByteDance Seed. 2025. Seed1.5-VL technical report. Technical report for the direct predecessor of the proprietary Doubao-Seed-1.6 model used in this work; no dedicated Seed 1.6 technical report is publicly available.

Mu Cai, Zeyi Huang, Yuheng Li, Utkarsh Ojha, Haohan Wang, and Yong Jae Lee. 2024. Leveraging large language models for scalable vector graphics-driven image understanding. *Preprint*, arXiv:2306.06094.

DeepSeek-AI. 2026. DeepSeek-V4: Towards highly efficient million-token context intelligence. Technical report, DeepSeek-AI. Released 24 April 2026; technical report at https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf.

Google DeepMind. 2025. Gemini. https://deepmind.google/models/gemini/.

Google DeepMind. 2026. Gemma 4: Byte for byte, the most capable open models. https://blog.google/innovation-and-ai/technology/developers-tools/gemma-4/. Released 2 April 2026; model card at https://ai.google.dev/gemma/docs/core/model_card_4.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1):335–346.

IBM Research. 2026. Introducing the IBM Granite 4.1 family of models. https://research.ibm.com/blog/granite-4-1-ai-foundation-models. Released 29 April 2026.

Barbara Landau, Linda B. Smith, and Susan S. Jones. 1988. The importance of shape in early lexical learning. *Cognitive Development*, 3(3):299–321.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, and 8 others. 2025. DeepSeek-V3 technical report. *Preprint*, arXiv:2412.19437.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, and 101 others. 2026. Ministral 3. *Preprint*, arXiv:2601.08584.

LLM-Core Xiaomi. 2025. MiMo: Unlocking the reasoning potential of language model – from pretraining to posttraining.

Boris Malashenko, Ivan Jarsky, and Valeria Efimova. 2025. Leveraging large language models for scalable vector graphics processing: A review. *Preprint*, arXiv:2503.04983.

MiniMax. 2026. Meet MiniMax-M2. https://github.com/MiniMax-AI/MiniMax-M2. Released 18 March 2026; building on the MiniMax-M2 base model (https://www.minimax.io/news/minimax-m2).

Mistral AI. 2026a. Introducing Mistral Small 4. https://mistral.ai/news/mistral-small-4. API model id mistral-small-2603; released 18 March 2026.

Mistral AI. 2026b. Mistral Medium 3.5: Remote agents in Vibe. https://mistral.ai/news/vibe-remote-agents-mistral-medium-3-5. API model id mistral-medium-3-5; released 29 April 2026.

Moonshot AI. 2026. Kimi k2.6: Advancing open-source coding. https://kimi.com/blog/kimi-k2-6. Released 20 April 2026; model card at https://huggingface.co/moonshotai/Kimi-K2.6.

NVIDIA. 2025. NVIDIA Nemotron 3: Efficient and open intelligence.

OpenAI. 2026a. Introducing GPT-5.4 mini and GPT-5.4 nano. https://openai.com/index/introducing-gpt-5-4-mini-and-nano/. Released 17 March 2026.

OpenAI. 2026b. Introducing GPT-5.5. https://openai.com/index/introducing-gpt-5-5/. Released 23 April 2026.

Zeju Qiu, Weiyang Liu, Haiwen Feng, Zhen Liu, Tim Z. Xiao, Katherine M. Collins, Joshua B. Tenenbaum, Adrian Weller, Michael J. Black, and Bernhard Schölkopf. 2025. Can large language models understand symbolic graphics programs? In *The Thirteenth International Conference on Learning Representations*.

Qwen Team. 2026a. Qwen3.5: Towards native multimodal agents. https://qwen.ai/blog?id=qwen3.5. Flagship Qwen3.5-397B-A17B released 16 February 2026; medium series 24 February; small series (incl. Qwen3.5-9B) 2 March 2026.

Qwen Team. 2026b. Qwen3.6. https://github.com/QwenLM/Qwen3.6. Released April 2026.

Pratyusha Sharma, Tamar Rott Shaham, Manel Baradad, Stephanie Fu, Adrian Rodriguez-Munoz, Shivam Duggal, Phillip Isola, and Antonio Torralba. 2024. A vision check-up for language models. *Preprint*, arXiv:2401.01862.

Anders Søgaard. 2023. Grounding the vector space of an octopus: Word meaning from raw text. *Minds Mach.*, 33(1):33–54.

V Team, Wenyi Hong, Xiaotao Gu, Ziyang Pan, Zhen Yang, Yuting Wang, Yue Wang, Yuanchang Yue, Yu Wang, Yanling Wang, Yan Wang, Xijun Liu, Wenmeng Yu, Weihan Wang, Wei Li, Shuaiqi Duan,

Sheng Yang, Ruiliang Lv, Mingdao Liu, and 78 others. 2026. Glm-5v-turbo: Toward a native foundation model for multimodal agents. *Preprint*, arXiv:2604.26752.

Ryan Teknium, Roger Jin, Jai Suphavadeeprasit, Dakota Mahan, Jeffrey Quesnelle, Joe Li, Chen Guang, Shannon Sands, and Karan Malhotra. 2025. Hermes 4 technical report. *Preprint*, arXiv:2508.18255.

xAI. 2025. Grok 4.1 Fast. https://x.ai/news. Released 19 November 2025.

xAI. 2026. Grok 4.3. https://docs.x.ai/. Beta released 17 April 2026; API general availability 6 May 2026.

Xiaomi. 2026. Introducing MiMo-V2.5. https://mimo.xiaomi.com/mimo-v2-5/. MiMo-V2 Flash released 16 December 2025; MiMo-V2 Omni 18 March 2026; MiMo-V2.5 22 April 2026; MiMo-V2.5 Pro 27 April 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, and 1 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Zhihui Yang, Yupei Wang, Kaijie Mo, Zhe Zhao, and Renfen Hu. 2025b. Does visual grounding enhance the understanding of embodied knowledge in large language models? *Preprint*, arXiv:2510.16924.

Z.ai. 2026. GLM-5: From vibe coding to agentic engineering. *Preprint*, arXiv:2602.15763. GLM-4.7-Flash released 19 January 2026; GLM-5 11 February 2026; GLM-5-Turbo 16 March 2026; GLM-5V-Turbo 1 April 2026; GLM-5.1 7 April 2026.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, Kedong Wang, Lucen Zhong, Mingdao Liu, Rui Lu, Shulin Cao, Xiaohan Zhang, Xuancheng Huang, and 1 others. 2025. GLM-4.5: Agentic, reasoning, and coding (ARC) foundation models. *Preprint*, arXiv:2508.06471.

Bocheng Zou, Mu Cai, Jianrui Zhang, and Yong Jae Lee. 2024. Vgbench: Evaluating large language models on vector graphics understanding and generation. *Preprint*, arXiv:2407.10972.

A Detailed results

Model	Acc. (%)	# non-parseable responses	# input tokens	# output tokens	Price in / out ($/M tok.)
google/gemini-3.1-pro-preview	94.5	0	230,495	92,266	$2.000 / $12.000
google/gemini-3-flash-preview	92.0	0	230,497	381,092	$0.500 / $3.000
openai/gpt-5.5	90.0	0	176,284	35,130	$5.000 / $30.000
google/gemini-3.5-flash	88.5	0	230,495	117,511	$1.500 / $9.000
moonshotai/kimi-k2.6	81.5	3	175,663	915,162	$0.740 / $3.490
anthropic/claude-sonnet-4.6	80.0	0	190,294	197,835	$3.000 / $15.000
qwen/qwen3.7-max	76.5	2	233,469	206,741	$2.500 / $7.500
x-ai/grok-4.3	76.0	1	199,227	675,972	$1.250 / $2.500
qwen/qwen3.5-397b-a17b	74.0	0	233,469	375,766	$0.390 / $2.340
qwen/qwen3.6-max-preview	74.0	0	233,469	250,545	$1.040 / $6.240
z-ai/glm-5-turbo	73.0	0	191,062	127,136	$1.200 / $4.000
google/gemma-4-31b-it	72.5	0	233,676	404,984	$0.130 / $0.380
deepseek/deepseek-v4-pro	71.5	13	176,664	408,241	$0.435 / $0.870
z-ai/glm-5v-turbo	71.0	1	191,062	183,180	$1.200 / $4.000
anthropic/claude-opus-4.7	70.5	0	244,866	2,686	$5.000 / $25.000
qwen/qwen3.6-plus	68.5	0	233,469	398,209	$0.325 / $1.950
xiaomi/mimo-v2.5-pro	68.5	0	275,738	130,754	$1.000 / $3.000
xiaomi/mimo-v2-omni	67.5	0	275,538	352,840	$0.400 / $2.000
openai/gpt-5.4-mini	67.0	0	176,284	80,721	$0.750 / $4.500
z-ai/glm-5.1	65.5	12	191,062	382,311	$1.050 / $3.500
deepseek/deepseek-v4-flash	65.0	0	179,798	448,393	$0.140 / $0.280
qwen/qwen3.6-27b	65.0	0	233,469	311,407	$0.320 / $3.200
qwen/qwen3.5-122b-a10b	64.5	0	233,469	464,869	$0.260 / $2.080
google/gemma-4-26b-a4b-it	63.0	1	233,644	573,210	$0.130 / $0.380
xiaomi/mimo-v2.5	63.0	3	274,938	208,355	$0.400 / $2.000
anthropic/claude-haiku-4.5	62.0	0	196,094	327,641	$1.000 / $5.000
bytedance-seed/seed-1.6	61.5	0	250,820	150,700	$0.250 / $2.000
x-ai/grok-4.1-fast	60.5	0	205,924	372,059	$0.200 / $0.500
nvidia/nemotron-3-super-120b-a12b	59.0	0	214,970	373,452	$0.090 / $0.450
qwen/qwen3.6-35b-a3b	58.5	0	233,469	397,097	$0.150 / $1.000
minimax/minimax-m2.7	58.0	0	183,886	686,919	$0.300 / $1.200
nvidia/llama-3.3-nemotron-super-49b-v1.5	53.5	0	179,038	327,681	$0.100 / $0.400
mistralai/mistral-medium-3-5	52.5	2	231,901	273,762	$1.500 / $7.500
openai/gpt-oss-20b	52.5	0	188,388	352,084	$0.003 / $0.140
mistralai/mistral-small-2603	52.0	0	231,901	604,856	$0.150 / $0.600
google/gemini-3.1-flash-lite-preview	51.5	0	230,590	600	$0.250 / $1.500
xiaomi/mimo-v2-flash	51.0	1	232,730	315,432	$0.090 / $0.290
nousresearch/hermes-4-405b	49.5	0	191,838	173,772	$1.000 / $3.000
nousresearch/hermes-4-70b	47.5	6	191,838	277,804	$0.130 / $0.400
openai/gpt-5.4-nano	47.5	0	176,284	52,875	$0.200 / $1.250
deepseek/deepseek-chat-v3-0324	47.0	0	179,647	4,004	$0.200 / $0.770
openai/gpt-oss-120b	46.0	9	187,955	83,747	$0.039 / $0.180
z-ai/glm-4.7-flash	45.0	3	191,062	554,539	$0.060 / $0.400
qwen/qwen3.5-9b	44.0	8	232,785	1,118,015	$0.100 / $0.150
bytedance-seed/seed-1.6-flash	40.5	7	251,220	217,574	$0.075 / $0.300
nvidia/nemotron-3-nano-30b-a3b	40.0	2	232,101	575,535	$0.050 / $0.200
mistralai/ministral-14b-2512	39.5	0	229,501	800	$0.200 / $0.200
mistralai/ministral-8b-2512	28.5	0	229,501	800	$0.150 / $0.150
mistralai/ministral-3b-2512	25.5	0	229,501	800	$0.100 / $0.100
ibm-granite/granite-4.1-8b	21.0	48	184,226	48,788	$0.050 / $0.100

Table 4: **Detailed per-model results on the 200-question benchmark.** "# non-parseable responses" counts outputs where the answer could not be extracted. Token counts are totals across all 200 questions. Prices are list prices per million tokens at evaluation time, shown as input / output.